

Software para la detección y el reconocimiento de rostros

Rafael Cazorla Martínez

Resumen—En los últimos años, las técnicas de detección y de reconocimiento facial han evolucionado mucho. Dado que este tipo de técnicas tienen muchas aplicaciones útiles en campos muy diversos como la biometría, la clasificación de imágenes o la seguridad, se han destinado muchos esfuerzos tanto económicos como científicos para tratar de mejorarlas.

En este trabajo se hará un estudio de las técnicas de detección y reconocimiento facial existentes en la actualidad. Se tratará de ver las diferencias existentes entre ellas y cómo son clasificadas. Tras ese primer análisis, se seleccionarán dos de las técnicas analizadas para su posterior implementación en un sistema capaz de detectar y reconocer rostros de personas introducidas previamente en el sistema. En este caso, se implementarán las técnicas Eigenfaces y Fisherfaces. Finalmente, se harán un conjunto de pruebas sobre diferentes bases de datos de imágenes para verificar y comparar los resultados obtenidos tras aplicar los diferentes algoritmos implementados.

Palabras clave—Detección facial, reconocimiento facial, eigenfaces, fisherfaces, pca, lda, extracción de características, normalización, biometría, características haar.

Abstract—In recent years, face detection and face recognition techniques have been improved. All these techniques have many useful applications and can be applied in many different areas such as biometrics, image classification or security. That's the reason why society has invested a lot of economic and investigation efforts in improving them.

In this paper, we will see the main face detection and face recognition techniques available at the moment. We will discuss the differences and how can we classify them depending on different features. Once we will have finished this first analysis, we will choose two of the analyzed techniques in order to develop them. We will develop a system able to detect and recognize faces from people who have been entered on the system previously in a training phase. In this case, the choosen techniques are Eigenfaces and Fisherfaces. Finally, we will make some tests on different image databases to analyze the results got from using the algorithms developed.

Index Terms—Face detection, face recognition, eigenfaces, fisherfaces, pca, lda, feature extraction, standardization, biometrics, haar features.



1 INTRODUCCIÓN

La reciente llegada y popularización de las redes sociales ha provocado que, durante los últimos años, se suban millones de fotografías a Internet a diario. El hecho de que estas fotografías se cuelguen en redes sociales ofrece la posibilidad de crear un archivo histórico digital de fotografías de personas. El problema está en que, pasado el tiempo, y dado el gran volumen de fotografías con las que se trabaja, es bastante fácil que olvidemos la identidad de las personas que aparecen. Las técnicas de detección y reconocimiento facial podrían tener la solución a este tipo de

problemas ya que consiguen reconocer la identidad de las personas que aparecen en las fotografías a partir de un entrenamiento previo. Este tipo de técnicas puede aplicarse tanto a las redes sociales como a otras muchas disciplinas como la seguridad, los sistemas de acceso, la clasificación de imágenes, la biometría forense o las aplicaciones móviles, entre muchas otras. Debido a esta gran cantidad de aplicaciones prácticas que ofrecen, en los últimos años se han centrado muchos esfuerzos en mejorar los algoritmos existentes e implementar nuevas técnicas, llegando a alcanzar grandes niveles de perfeccionamiento.

El proyecto consistirá en la implementación de un sistema de detección y reconocimiento facial que constará de varias fases. En primer lugar una primera fase de búsqueda de información dónde se podrá ver cual es el estado del arte y las técnicas más avanzadas que existen en la ac-

- E-mail de contacto: rafael.cazorlam@e-campus.uab.cat
- Mención realizada: *Tecnologies de la Informació*.
- Trabajo tutorizado por: Joan Serra Sagristà (dEIC)
- Curso 2015/16

tualidad. A continuación, se verá en qué consisten las principales técnicas así como una clasificación de éstas. En este punto se describirán también las fases que tiene un sistema de reconocimiento facial. Una vez se han visto las técnicas existentes, se seleccionará una técnica de detección y dos de reconocimiento para su implementación en Python. Finalmente se realizará un conjunto de pruebas con las diferentes técnicas implementadas para analizar y verificar los resultados obtenidos.

1.1 Objetivos

La motivación que me ha llevado a la selección de este proyecto es la curiosidad por entender el funcionamiento de los sistemas de reconocimiento facial, cada vez más presentes en nuestro día a día. No hay mejor manera para comprender el funcionamiento de un sistema que llevar a cabo su implementación.

Así pues, podemos dividir los objetivos del proyecto y listarlos según su prioridad, como en la Tabla I.

TABLA I.
LISTADO DE OBJETIVOS DEL PROYECTO

Objetivo	Descripción	Prioridad
O1	Documentarse sobre las diferentes técnicas existentes.	Prioritario
O2	Desarrollar un software de detección de caras.	Prioritario
O3	Desarrollar un software de reconocimiento de caras.	Prioritario
O4	Conseguir un porcentaje de acierto en el reconocimiento superior al 60%.	Secundario
O5	Realizar el proyecto en el tiempo especificado.	Crítico

Fig. 1. Listado de objetivos de Software para la detección y el reconocimiento de rostros.

El primer objetivo (O1) establece la necesidad de documentarse para conocer en qué consisten los sistemas de detección y reconocimiento facial y conocer las diferentes técnicas. De este modo se podrá hacer una selección adecuada que permita llevar a cabo el resto de objetivos en el tiempo establecido.

El segundo objetivo (O2) implica el desarrollo de un software capaz de detectar los rostros que aparecen en una imagen.

El tercer objetivo (O3), parecido al anterior, implica el desarrollo de un software capaz de reconocer en una imagen los rostros de personas conocidas por el sistema. Así pues, será necesario que el sistema haya sido entrenado previamente con imágenes del individuo que se quiere reconocer. Esta información deberá ser guardada en un archivo externo o base de datos para que pueda ser cargada posteriormente.

En cuanto al cuarto objetivo (O4), consiste en la mejora del sistema para obtener resultados en el reconocimiento con

porcentajes de acierto superior al 60%. Este objetivo es secundario ya que se priorizará el hecho de obtener un sistema que funcione correctamente y, una vez conseguido, se tratarán de mejorar los resultados.

El último objetivo (O5) y el más prioritario es llevar a cabo la totalidad del proyecto y conseguir el resto de objetivos dentro del tiempo especificado.

1.2 Contenido

En primer lugar se hace una introducción al estado del arte en la sección II. En la sección III se explican las fases de un sistema de reconocimiento facial y se propone una posible forma de clasificación de las técnicas de reconocimiento. La metodología y la planificación seguidas se muestran en la sección IV. A continuación, en las secciones V y VI, se explica el diseño de la aplicación y los detalles de la implementación llevada a cabo. Por último en las secciones VII y VIII se pueden ver un conjunto de pruebas para evaluar el rendimiento de la aplicación y las conclusiones que se pueden extraer del trabajo.

2 ESTADO DEL ARTE

El reconocimiento facial automatizado es un concepto relativamente nuevo. Desarrollado en los años 60, el primer sistema para reconocimiento facial era semiautomático y requería un administrador para localizar rasgos en las fotografías antes de que este calculara distancias a puntos de referencia en común. No fue hasta finales de los años 80 que, con la aparición de técnicas como el Principal Component Analysis (PCA)[1], se crearon los primeros sistemas automatizados de reconocimiento facial en tiempo real fidedignos. A partir de ese punto, surgieron dos enfoques predominantes: el fotométrico o basado en apariencia, que analiza una imagen en plano, y el geométrico o basado en modelos que analiza las características de los rasgos (ojos, nariz, etc.).

A medida que han ido pasando los años, el interés por este campo ha ido aumentando progresivamente hasta el día de hoy. En la actualidad, las nuevas técnicas de reconocimiento facial han conseguido niveles de perfeccionamiento muy altos.

Existen muchas compañías compitiendo por conseguir el mejor algoritmo de reconocimiento facial. Su objetivo es conseguir un nivel de precisión prácticamente igual al de los humanos. Dos de las principales empresas tecnológicas del momento, disponen de los algoritmos más avanzados hoy en día. Ellas son Facebook y Google.

En 2014, Facebook anunció que había conseguido obtener un 97,3% de porcentaje de acierto con su algoritmo DeepFace[2] probado en la base de datos de imágenes LFW (Labeled Face in the Wild). Ese resultado mejoraba en un 27% el mejor resultado hasta el momento. Google, por su parte, con su algoritmo FaceNet[3] obtenía un porcentaje superior con un 99,6% de aciertos.

En Enero de 2015, Facebook presenta un nuevo método de reconocimiento llamado PIPER[4] capaz de reconocer a

personas en poses inusuales, incluso si su cara no es visible. Este algoritmo ya no es solamente capaz de fijarse en la cara, sino que además analiza aspectos como el peinado, la ropa o los complementos.

Así pues, el reconocimiento facial, entendido como método para identificar la identidad de una persona, evoluciona hacia nuevos niveles. Ahora existen algoritmos capaces de identificar a personas por características muy diferentes y no solo por su rostro.

No obstante, entre los objetivos de este trabajo no se encuentra la creación de nuevos algoritmos o formas de reconocimiento facial sino el estudio e implementación de algunas técnicas ya existentes.

3 FASES DE UN SISTEMA DE RECONOCIMIENTO

Un sistema de reconocimiento facial, como se ve en la Fig. 2, consta normalmente de 4 fases. De estas 4 fases como mínimo 3 de ellas son totalmente necesarias. Existe una cuarta, la de normalización, que no es imprescindible pero sí muy recomendable para optimizar los resultados.

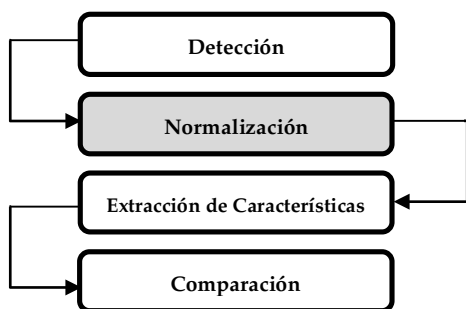


Fig. 2. Diagrama representativo de las 4 fases que conforman un sistema de reconocimiento facial.

La fase de detección consiste en encontrar áreas de la imagen que contengan una cara para aislarlas del resto. Este es uno de los objetivos del proyecto, y una de las fases más importante de la aplicación ya que una mala detección conllevaría un error en el resto de las fases. Como se verá en la implementación, también es importante detectar la posición de los ojos.

La fase de normalización es importante si se quieren mejorar los resultados. Esta fase consiste en la localización de los componentes principales de la cara y, mediante transformaciones geométricas, normalizar aspectos como el tamaño, la posición de la cara o las condiciones de luminosidad.

La fase de extracción de características puede variar según la técnica empleada pero mantiene en todos los casos una misma finalidad. En esta fase, el sistema extrae una serie de características propias de un rostro y las guarda en una base de datos.

En la última fase, se compara la información extraída del rostro que se quiere reconocer con los datos que hay almacenados en la base de datos. Si al comparar se encuentra un rostro con un porcentaje elevado de similitud, el sistema devuelve la identidad de la cara.

3.1 Técnicas de reconocimiento facial

Durante las últimas décadas se han desarrollado un gran número de algoritmos para el reconocimiento facial. Según[5], estos algoritmos se pueden clasificar en dos grandes grupos. En primer lugar las técnicas basadas en apariencia, que analizan la textura de la imagen a partir de la cual se aplicarán diferentes técnicas estadísticas y se extraerá la información. En segundo lugar, las técnicas basadas en modelos. Éstas, por su parte, extraen las características tanto de la forma del rostro como de la textura de la cara.

Dentro de cada una de estas clasificaciones, encontramos distintos sub apartados. En la Fig. 3 se muestra de forma más esquemática la clasificación de las técnicas.

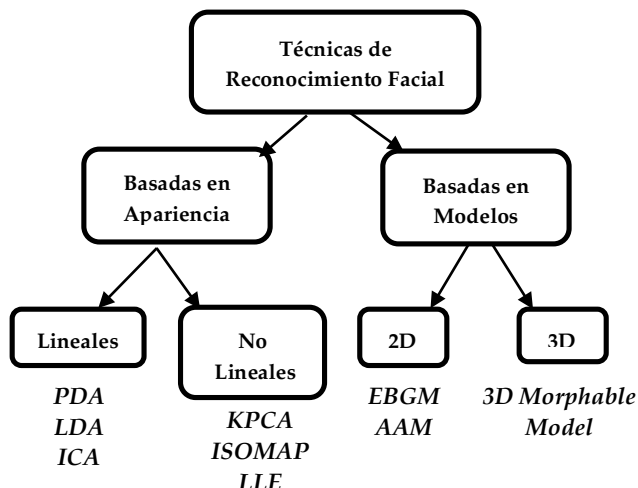


Fig. 3. Diagrama que muestra una propuesta de taxonomía para las técnicas de reconocimiento facial.

A. Técnicas basadas en apariencia

Las técnicas basadas en apariencia transforman el problema de reconocimiento facial en un problema de análisis de espacio donde se pueden aplicar diferentes técnicas estadísticas. De este tipo de técnicas destaca su aplicabilidad en imágenes de baja resolución o mala calidad, su rapidez de ejecución (se pueden implantar en sistemas en tiempo real) o su baja complejidad. Sin embargo también tienen varios inconvenientes. Uno de los inconvenientes es que para conseguir buenos resultados se requiere un conjunto de muestras considerable para la fase de entrenamiento. También aspectos como los cambios en la iluminación, la pose o la expresión de la cara tienen un gran impacto en los resultados finales. Dependiendo del método empleado, estos inconvenientes tendrán un impacto mayor o menor.

Dentro de las técnicas basadas en apariencia se encuentran las técnicas lineales (con algoritmos como el PCA, el LDA o el ICA) y las no lineales (con el KPCA, el ISOMAP o LLE).

B. Técnicas basadas en modelos

Las técnicas basadas en modelos, tratan de obtener características biométricas de las imágenes para realizar el reconocimiento. Se tienen en cuenta aspectos como la distancia entre los ojos, el grosor de la nariz, el tamaño de la boca, etc. Estos sistemas requieren un conocimiento previo de las imágenes, además de ser más lentos y complejos que los sistemas basados en apariencia. Sin embargo, son más robustos frente a cambio de orientación o expresión de la cara y se ven menos afectados por cambios en la iluminación o las sombras. Dentro de las técnicas basadas en modelos, nos encontramos con dos divisiones: las técnicas 2D y las 3D.

Las técnicas 2D ofrecen una descripción precisa de las diferentes partes que componen la cara (ojos, nariz, boca...) y diferentes propiedades como la distancia que hay entre las partes. Las técnicas que utilizan imágenes en 3D, en cambio, ofrecen información relativa a la forma, la profundidad y la textura de la cara. Así, las técnicas 3D mejoran los resultados de las técnicas 2D además de poder reconocer caras desde distintos ángulos o perspectivas. Sin embargo, el tiempo de cómputo es superior y en muchas ocasiones se requiere de instrumental de captación especializado.

En la Fig. 4 podemos ver representada la extracción de características de un rostro tras la aplicación de una técnica basada en modelos 2D y 3D respectivamente.



Fig. 4. A la izquierda, ejemplo de técnica 2D. A la derecha, resultado tras la modelización de un rostro con técnicas 3D.

4 METODOLOGIA

La metodología elegida para este proyecto es la clásica y más popular: el modelo en cascada[6]. Esta metodología ordena las etapas rigurosamente de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. No obstante, se contemplará en todo momento una revisión al final de cada fase de tal forma que si se detecta un error se podrá volver a la fase anterior para corregirlo.

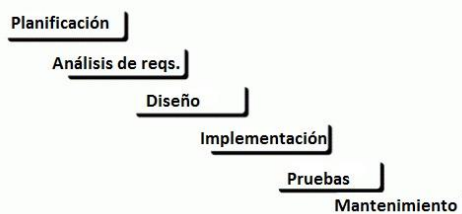


Fig. 5. Modelo en cascada: diagrama que muestra las fases de la metodología seguida en el proyecto.

Como se puede ver en Fig.5, esta metodología consta de las siguientes 6 fases:

1. Planificación o análisis: En esta primera fase se realiza un estudio de la aplicación a desarrollar y se planifican las tareas asociadas para llevarla a cabo.

2. Análisis de requerimientos: Consiste en identificar las características y funcionalidades de la aplicación, teniendo en cuenta los objetivos que debe cumplir.

3. Diseño: En esta fase se define el diseño y la estructura de la solución final (módulos, funciones y sus relaciones). También se definen los algoritmos necesarios.

4. Implementación: En esta fase se traduce el diseño de la fase anterior a código de tal forma que obtenemos una aplicación funcional.

5. Pruebas: Se comprueba que los elementos programados funcionan correctamente y que cumplen con los requisitos establecidos en la fase 2.

6. Mantenimiento: Consiste en dar soporte y ofrecer mejoras al software una vez ha sido entregado al usuario final. En este caso, puesto que la aplicación no tiene un uso comercial, no será necesaria.

Se trata de un proyecto corto, realizado por una sola persona, y con los objetivos bien definidos. Además los requerimientos son claros y no existe ningún usuario externo para el que se desarrolle el proyecto, por lo que no se prevén cambios inesperados en los requisitos. Por estas razones, se considera que la metodología en cascada es la que más se ajusta al proyecto dada su sencillez y su viabilidad según las condiciones descritas anteriormente.

4.1 Planificación

El objetivo de este punto es definir y preparar las condiciones de trabajo que se llevarán a cabo. Para ello se establecen unas fechas límite para cada una de las fases de tal forma que se puedan alcanzar los objetivos del proyecto en la fecha prevista. El proyecto se ha planificado teniendo en cuenta las siguientes fases:

1. Planificación: (Semana 1 - 3) Del 15/2 al 13/3.

2. Diseño: (Semana 4 - 6) Del 14/3 al 27/3.

3. Implementación: (Semana 7 - 13) Del 28/3 al 22/5.

4. Pruebas: (Semana 14 - 16) Del 23/5 al 12/6.

5. Entrega final: (Semana 17 - 18) Del 13/6 al 26/6.

6. Defensa proyecto: (Semana 19-20) Del 27/6 al 7/7.

Cada fase lleva un conjunto de tareas asociadas que pueden estar marcadas por sus dependencias.

TABLA II.
FASES Y ACTIVIDADES DEL PROYECTO

	Nombre de la fase	Tareas asociadas
1	Planificación	- Búsqueda de información - Definición de objetivos y requisitos
2	Diseño	- Búsqueda de información y técnicas - Selección de las técnicas - Diseño del software
3	Implementación	- Desarrollo del software
4	Pruebas	- Pruebas de unidad - Pruebas rendimiento aplicación
5	Entrega final	- Redacción de documentos
6	Defensa del proyecto	- Preparación de la presentación - Presentación

Fig. 6. Fases del proyecto y tareas asociadas a cada fase.

La duración del proyecto será de 21 semanas o, lo que es lo mismo, 167 días. Con fecha de inicio el 15 de Febrero de 2016 y fecha final el 7 de Julio de 2016.

5 DESCRIPCIÓN Y DISEÑO DE LA APLICACIÓN

Tras haber llevado a cabo la fase de búsqueda de información, es necesario decidir qué técnicas van a desarrollarse. Para tomar la decisión se tiene en cuenta un factor determinante: el tiempo. Como se puede ver en Fig. 1, el quinto objetivo (O5) es el de mayor prioridad y requiere acabar el proyecto dentro del tiempo especificado. Por este motivo, se decide implementar las técnicas de reconocimiento basadas en PCA y LDA que, aplicadas al ámbito de reconocimiento facial son también conocidas como Eigenfaces y Fisherfaces respectivamente. La cuestión es que, pese a ser técnicas diferentes, gran parte de su implementación es muy similar de tal forma que se puede reducir el tiempo que sería necesario para desarrollar dos técnicas que fueran completamente distintas.

En cuanto a la técnica de detección de rostros y de ojos utilizada, se ha seleccionado la detección basada en haar-like features o características haar. Aparte de ser la técnica más extendida, su implementación es relativamente sencilla y ofrece buenos resultados.

5.1 Descripción del entorno de desarrollo

En este punto se describe el entorno de desarrollo de la aplicación.

A. Lenguaje de programación

El lenguaje de programación elegido para la implementación es Python 3.5.1[7], la última versión estable. Python es un lenguaje de programación orientado a objetos muy utilizado en el desarrollo de algoritmos de inteligencia artificial, por lo que existen diferentes librerías y herramientas que pueden ser muy útiles para conseguir los objetivos de una manera más sencilla.

B. Entorno de trabajo

El entorno de trabajo seleccionado es PyCharm, un IDE desarrollado por JetBrains que proporciona análisis de código, depuración gráfica, entre otras muchas características. La versión utilizada es PyCharm Community Edition 5.0.4, una versión gratuita.

C. Librerías y extensiones

Para el desarrollo de la aplicación se hace uso de diferentes librerías que nos facilitarán la implementación de los algoritmos descritos. Las principales son las siguientes:

OpenCV[8] es una librería Open Source desarrollada para resolver problemas de visión por computador. Dispone de más de 500 funcionalidades, entre las que destacan la detección y el reconocimiento de objetos.

NumPy[9] es una librería Open Source de Python que ofrece un conjunto de funciones matemáticas para operar más fácilmente con vectores y matrices.

5.2 Descripción del funcionamiento del software

Tal y como sucede en la mayoría de sistemas de reconocimiento facial, el sistema que se implementa en este proyecto consta de 4 fases. Cada una de estas fases realiza un conjunto de funcionalidades diferentes, desde la detección de caras hasta la decisión final, pasando por las fases de normalización y la de extracción de características.

Existe una primera fase de entrenamiento del sistema. En esta fase, se procesan un conjunto de imágenes de muestra de un individuo con el fin de guardar sus características en una base de datos. Este proceso se repite con los diferentes individuos que queramos poder reconocer.

Una vez se ha entrenado al sistema añadiendo individuos a la base de datos, podemos pasarle una imagen para que detecte y reconozca los individuos que aparecen, si éstos se encuentran ya en la base de datos.

5.3 Diseño de la aplicación

Una fase muy importante previa a la codificación es el diseño de la aplicación. Esta es la primera fase del proyecto que se puede considerar dentro del desarrollo software, ya que las fases anteriores estaban más relacionadas con la documentación y la búsqueda de información. Durante esta fase, se crea un modelo arquitectónico y funcional de la aplicación que posteriormente será implementada. De esta forma, sentamos las bases y la estructura principal de nuestra aplicación final.

En el diseño de esta aplicación se han hecho diferentes tipos de diagramas UML que permiten determinar la estructura y el funcionamiento del software.

Se ha desarrollado un diagrama de clases dónde se muestran las diferentes clases existentes en el sistema así como las relaciones entre ellas. También se han definido varios diagramas de secuencia de funciones que el usuario puede llevar a cabo con la base de datos.

6 DESARROLLO DEL SOFTWARE

En este punto se verá con detalle los pasos seguidos para llevar a cabo la implementación. La explicación está dividida según las diferentes fases del sistema de reconocimiento facial.

6.1 Detección

Es la primera fase del sistema. En esta fase, se realiza la detección tanto del rostro como de los ojos, en ese orden. El orden es importante ya que se detectará primero la región de la cara y, a continuación, se buscará solamente dentro de esta región la localización de los ojos. De esta forma se consigue reducir el tiempo de ejecución del algoritmo ya que la búsqueda de ojos se lleva a cabo dentro de un área más reducida.

A. Detección de rostro

Esta es la primera acción que se lleva a cabo con la imagen cuando se recibe una muestra externa.

Para la detección de rostros se han utilizado haar-like features bajo el algoritmo de Viola and Jones[10]. Las haar-like features aprovechan el conocimiento extraído de la observación de un gran número de caras. Normalmente, la zona de los ojos suele ser siempre más oscura que la zona de las mejillas o de la nariz. Como se puede ver en la Fig. 7, está información en forma de Haar es utilizada para llevar a cabo la detección.

El algoritmo de Viola and Jones, además, permite la detección de caras en tiempo real dado su bajo coste computacional.

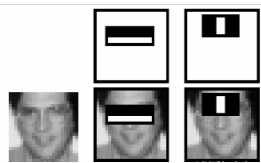


Fig. 7. Ejemplo de clasificador Haar para la detección de rostros.

La implementación se ha hecho con la librería OpenCV[8], que incorpora la funcionalidad de detectar objetos mediante Viola and Jones. En el proyecto se utiliza un clasificador Haar para la detección frontal de rostros que ya incluye la librería OpenCV.

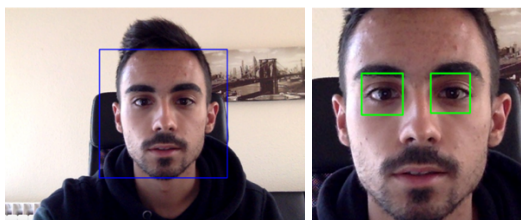


Fig. 8. Detección de rostro y de ojos utilizando un clasificador Haar.

B. Detección de ojos

El proceso de detección de ojos es muy similar al de detección de rostros. No obstante, este punto adquiere más importancia en la fase de normalización ya que, conociendo la posición de los ojos, se puede llegar a determinar el ángulo de rotación de la cara para alinearla.

A la hora de detectar no se hace distinción entre ojo derecho y ojo izquierdo, por ello se puede utilizar un clasificador haar genérico para ambos ojos. En la Fig. 8 podemos ver el resultado de aplicar el proceso de detección de ojos sobre la imagen de un rostro.

6.2 Normalización

En esta segunda fase, se le aplican una serie de transformaciones a las imágenes de entrada para que la extracción de características sea lo más efectiva posible.

Las operaciones que se llevan a cabo en esta fase son la rotación, el escalado, el recorte y la ecualización del histograma. Tras finalizar esta fase se obtiene una imagen en la que aparece únicamente el rostro ecualizado.

A. Rotación

La primera transformación que se aplica es la rotación. Para ello, se requiere la posición de los ojos que se ha obtenido anteriormente en la fase de detección. También es necesario saber el punto central del rostro para tomarlo como eje de rotación. Mediante un cálculo trigonométrico, se puede obtener el ángulo de rotación necesario, respecto al eje horizontal, para alinear verticalmente el rostro. En la Fig. 9, podemos ver los resultados de llevar a cabo esta transformación.

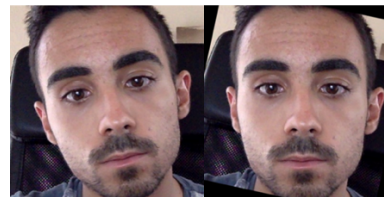


Fig. 9. Proceso de rotación y alineación del rostro. A la derecha el resultado obtenido de la alineación del rostro de la izquierda.

B. Escalado

El escalado consiste en proporcionar un tamaño determinado al rostro procesado de forma que todos tengan las mismas proporciones. Para realizar esta operación existen diversas alternativas, en esta aplicación se ha optado por seguir el estándar propuesto por la norma ISO/IEC 19794-5[11] que aconseja que la distancia entre el centro de los dos ojos sea como máximo de 96 píxeles.

Tomando esa distancia como referencia para el proceso de escalado, se consigue que todas las imágenes tengan unas proporciones muy similares facilitando, así, las tareas de comparación.

C. Recorte

Después del proceso de escalado, es el momento de recortar la imagen. Se le da a todas las imágenes las mismas dimensiones finales. Del mismo modo que en el punto anterior, se sigue el estándar que marca la norma en [11]. Así, las fotografías finales tienen unas dimensiones de 168 x 192 píxeles. (ancho x alto.)

En la Fig. 10 se muestra el resultado de aplicar estas dos últimas transformaciones a diferentes imágenes.

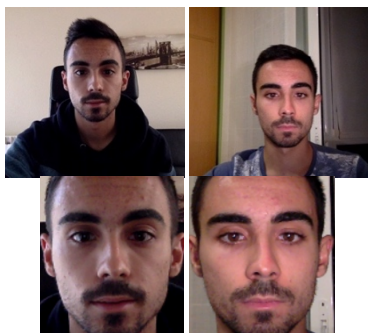


Fig. 10. Resultado de aplicar el escalado y el recorte sobre dos imágenes diferentes. Las dimensiones de las imágenes finales no corresponden a las dimensiones originales ya que ha sido necesario adaptarlas al formato del documento.

D. Ecualización del histograma

Por último, para acabar con el proceso de normalización, se realiza una ecualización del histograma. El objetivo de este proceso es conseguir que el histograma de la imagen sea lo más uniforme posible, es decir, que el número de píxeles para cada nivel de gris (0 a 255) sea lo más igualado posible. Con este proceso se consigue mitigar los cambios de luminosidad que afectan negativamente a los algoritmos de extracción de características que se utilizarán, concretamente, al PCA.

Para poder realizar la ecualización, primero se pasa la imagen en color a escala de grises y posteriormente se ecualiza el histograma[12]. Existen funciones en OpenCV que permiten hacerlo fácilmente.



Fig. 11. La ecualización del histograma permite compensar los cambios de luminosidad para hacer más efectiva la extracción de características. A la izquierda imagen original oscura, a la derecha imagen ecualizada.

6.3 Extracción de características

En la fase de extracción de características se extraen una serie de características propias de un rostro para posteriormente clasificarse, analizarse y compararse. En este punto contamos con imágenes ya normalizadas.

Mediante la aplicación de los dos algoritmos que se han elegido, se obtienen los valores que realmente aportan información sobre una cara y nos desprendemos de aquellos que no aportan información o pueden ser confusos de cara al reconocimiento en la fase de comparación.

El objetivo de estos algoritmos es clasificar las diferentes caras en un nuevo subespacio. Dado el conjunto de muestras de imágenes de entrenamiento de un individuo, éstas se proyectan en un nuevo subespacio, como se muestra en la Fig. 12.

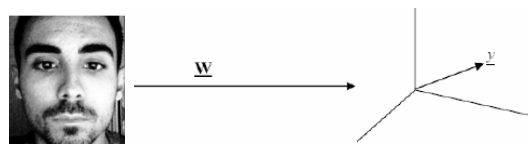


Fig. 12. Esquema que muestra la proyección de una imagen al nuevo subespacio.

A. PCA (Principal Component Analysis)

La primera técnica implementada es el PCA[1][13]. Su objetivo es reducir la información de las muestras del conjunto de entrenamiento, manteniendo únicamente un conjunto de componentes principales capaces de representar mejor los datos en forma de mínimos cuadrados. Durante la fase de entrenamiento, aplicando el algoritmo PCA, se obtienen un conjunto de valores y vectores propios. Estos valores y vectores propios se ordenan en orden decreciente de forma que los primeros serán los que más información relevante contienen, mientras que los últimos son los que aportan menos información.

A grandes rasgos, la secuencia que se sigue cuando se quiere reconocer un rostro en una imagen es la siguiente:

1) Transformación de la muestra para que pueda ser proyectada en el sub-espacio PCA.

2) Proyección en el sub-espacio PCA mediante la multiplicación del vector fila por la matriz de transformación PCA calculada previamente en la fase de entrenamiento. La matriz corresponde al conjunto de vectores propios y al realizar la multiplicación obtenemos un vector columna.

B. LDA (Linear Discriminant Analysis)

La técnica de Análisis Discriminante Lineal[14] se considera una técnica derivada de PCA y, de la misma forma que ésta, su uso está también muy extendido. Esta técnica pretende maximizar la varianza entre diferentes clases y minimizar la varianza dentro de la propia clase. Para ello, LDA[15] proyecta los datos (imágenes) en un espacio vectorial de baja dimensión de manera que el ratio de la distancia entre clases y la distancia dentro de la misma clase se maximiza.

En cuanto a resultados, teóricamente este algoritmo ofrece un mejor rendimiento ante los cambios de iluminación en las imágenes. Problema que, por otro lado, ya se ha intentado mitigar en la fase de normalización.

6.4 Comparación

Finalizada la extracción de características, se llega a la última fase. En esta fase se compara la información obtenida con la que ya existía. Para calcular la distancia se usa la distancia Euclídea o Euclidiana con el fin de tomar una decisión.

En un plano, la distancia euclidiana es la existente entre dos puntos P_1 y P_2 , de coordenadas cartesianas (x_1, y_1) y (x_2, y_2) respectivamente, como se muestra en (1):

$$d_e(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Teniendo en cuenta la explicación relativa a la Fig. 12, en la fase de comparación se analiza la distancia entre el vector de la imagen de muestra donde se quiere hacer el reconocimiento y los vectores correspondientes a las imágenes de entrenamiento. Se busca, pues, el vector de las imágenes de entrenamiento que más cerca esté del vector analizado, como en la Fig. 13. Este vector corresponderá al individuo que más similitud presente con la muestra analizada.

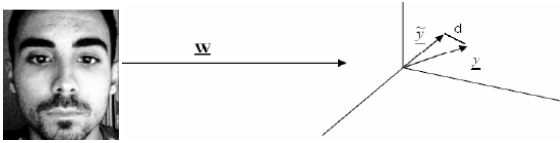


Fig. 13. Esquema que muestra la comparación de distancias entre dos vectores en el nuevo subespacio donde se representan las imágenes.

7 FASE DE PRUEBAS Y RESULTADOS

El objetivo de esta fase es evaluar el rendimiento que ofrece la aplicación. Para llevar a cabo el conjunto de pruebas se utilizarán principalmente dos bases de datos: la *ORL Face Database*[16] y la *extended Yale B face database Cropped*[17][18].

A. ORL Face Database

Es la base de datos de AT&T. Contiene 40 individuos con 10 imágenes para cada uno, haciendo un total de 400 imágenes. Como se puede ver en la Fig. 14, los individuos aparecen de frente pero con diferentes poses y expresiones faciales. Además, algunos sujetos llevan complementos como barba o gafas. La dimensión de las imágenes no es igual en todos los casos por lo que será necesario adaptarlo para la extracción de características.



Fig. 14. Muestras de un individuo de la base de datos ORL y de la Yale B Face Database Cropped respectivamente.

B. Extended Yale B Face Database Cropped

Es la base de datos de la universidad de Yale. Contiene 2400 imágenes de 38 individuos. Cada individuo aparece en 63 imágenes con diferentes condiciones de luminosidad y exposición a la luz. Un ejemplo de los cambios de iluminación sobre un mismo individuo aparece en la Fig. 14. En

este caso, el conjunto de imágenes utilizado ya ha sido normalizado por lo que todas las imágenes están alineadas y tienen las mismas dimensiones.

7.1 Pruebas de detección

En esta primera parte de las pruebas se evaluará el sistema de detección de rostros y de ojos incorporado en la aplicación de reconocimiento facial. Las pruebas se hacen sobre imágenes elegidas aleatoriamente y donde aparecen uno o más rostros.

TABLA III.

DETECCIÓN DE ROSTROS Y OJOS EN UN CONJUNTO DE 32 IMÁGENES

	Aciertos	Error detección de ojos	Detección de ojos correcta
Detección rostro	28 / 32	9 / 28	18 / 28
Porcentaje	87,5%	32,14%	64,28%

Fig. 15. Resultados de la detección de rostros y ojos en un conjunto de 32 imágenes. Resultados de detección de ojos relativos a la detección de rostros.

Como se puede ver en la Tabla III, el porcentaje de detección de rostros que ofrece el sistema implementado es alto. No obstante, como ya se explicó anteriormente, para llevar a cabo el proceso de normalización y que sea efectivo se requiere de la correcta detección de los dos ojos. Si se tiene en cuenta este aspecto, en el caso de la Tabla III, el porcentaje de acierto total se vería limitado no solo a la detección de rostros sino también a la detección de ojos. De esta forma el porcentaje de acierto final sería: $18/32 = 56,25\%$. Muy inferior al 87,5 que se consigue con la simple detección de rostro.

7.2 Pruebas de reconocimiento en imágenes con variación en la pose y la expresión

En este caso se analiza cómo afectan los cambios en la pose y en la expresión facial al rendimiento de los algoritmos implementados. Para ello se evalúa el porcentaje de acierto sobre la base de datos de ORL con diferente número de componentes principales. Como ya se vio, los componentes principales tienen la información más importante de un rostro y están ordenados según la cantidad de información que contienen. Por este motivo, el primer componente será el que más información aporte y a medida que vamos aumentando el número de componentes estos aportarán cada vez menos información de manera que los últimos componentes no tendrán ningún efecto y podrían ahorrarse.

Las pruebas de la Tabla IV se realizan con todas las imágenes(400) de la base de datos ORL. Como se puede ver, los primeros 10 componentes son los que más información aportan para Eigenfaces ya que el porcentaje de acierto con un número de componentes superior a diez no cambia significativamente. En el caso de Fisherfaces es más visible aún. El primer componentes obtiene un porcentaje de acierto de más del 99%.

De la Tabla IV también se puede extraer que Fisherfaces tiene un rendimiento superior a Eigenfaces bajo condiciones de variación en la pose y la expresión. Esta diferencia se hace más notable si el número de componentes es bajo.

TABLA IV.
 RESULTADO DE LAS PRUEBAS EN LA BASE DE DATOS ORL

#	Eigenfaces		Fisherfaces	
Componentes	%acierto	%error	%acierto	%error
1	9,06	90,94	99,75	0,25
5	89,21	10,79	100	0
10	96,07	3,93	100	0
15	97,79	2,21	100	0
20	97,79	2,21	100	0
40	98,53	1,47	100	0
80/40*	98,28	1,72	100	0
100/40*	98,28	1,72	100	0

Fig. 16. Resultado de las pruebas en la base de datos de ORL cropped en términos de acierto y error porcentuales. *El número máximo de componentes para Fisherfaces es de 40, igual al número de individuos (clases).

7.3 Pruebas de reconocimiento en imágenes con variación en las condiciones de iluminación

En este segundo caso se analizan los efectos que tienen los cambios de luminosidad sobre el rendimiento de los algoritmos implementados. Para ello se evalúa el porcentaje de acierto sobre la base de datos de *Yale B Cropped* con diferente número de componentes principales.

Como conocimiento previo, se sabe que los cambios de luminosidad normalmente afectan más aplicando el algoritmo Eigenfaces que Fisherfaces.

En la Tabla V se muestran los resultados obtenidos. Se corrobora que el rendimiento de Eigenfaces se ve altamente afectado por los cambios de luminosidad ya que, comparándolo con los resultados de la Tabla IV, para todo número de componentes el porcentaje de acierto es muy inferior. Además los valores más altos que alcanza están alrededor del 75%, muy lejos de los que se obtuvieron en la prueba anterior.

TABLA V.
 RESULTADO DE LAS PRUEBAS EN LA BASE DE DATOS YALE B CROPPED

#	Eigenfaces		Fisherfaces	
Componentes	%acierto	%error	%acierto	%error
1	4,68	95,32	99,79	0,21
5	4,8	95,2	100	0
10	20,03	79,97	99,99	0,01
15	32,53	67,47	100	0
20	42,03	57,97	99,24	0,76
40/38*	57,55	42,45	99,12	0,88
80/38*	69,46	30,54	99,12	0,88
100/38*	72,64	27,36	99,12	0,88

Fig. 17. Resultado de las pruebas en la base de datos de Yale B Cropped en términos de acierto y error porcentuales. *El número máximo de componentes para Fisherfaces es de 38, igual al número de individuos (clases).

Por otro lado, cabría destacar que aunque los resultados obtenidos para Fisherfaces tanto en la Tabla IV como en la Tabla V son muy buenos, no se comporta así en todas las bases de datos. En [20], por ejemplo, se muestran los resultados de aplicar Fisherfaces sobre la base de datos LFW. En ese caso, el porcentaje de acierto se encuentra alrededor del 65%.

7.4 Comparativa con otras técnicas

Tras haber visto los resultados obtenidos por la aplicación implementada, es momento de comparar estos resultados con otras técnicas. Como se ha visto anteriormente, las técnicas implementadas (Eigenfaces y Fisherfaces) son técnicas basadas en apariencia que transforman el problema de reconocimiento facial en un problema de análisis de espacio mediante técnicas estadísticas. En este punto, se busca comparar los resultados con otro tipo algoritmos como por ejemplo las técnicas basadas en modelo.

La comparación se lleva a cabo con uno de las técnicas más populares dentro de las técnicas basadas en modelos: el Elastic Bunch Graph Matching (EBGM)[19].

Para que la comparación sea óptima, es necesario que las pruebas se hayan efectuado bajo las mismas bases de datos. En [20] se pueden encontrar los resultados tras aplicar EBGM en las bases de datos de Yale y ORL. En la tabla siguiente se muestra la comparativa entre las tres técnicas (PCA, LDA, EBGM).

Cabe decir que existen diferentes implementaciones de EGBM que ofrecen resultados muy diferentes por lo que la comparación y las conclusiones que se saquen de este punto no serán aplicables a todos los algoritmos de EGBM.

TABLA VI.
 COMPARATIVA CON OTROS ALGORITMOS EN BASE DE DATOS ORL

	Eigenfaces		Fisherfaces		EBGM	
%	acierto	error	acierto	error	acierto	error
Mejor	98,53	1,47	100	0	91,5	8,5

Fig. 18. Comparativa de resultados con técnica basada en modelo (EBGM). Pruebas hechas en la base de datos de ORL y resultados medidos en términos de acierto y error porcentuales.

En la Tabla VI se puede ver que ante los cambios en la pose y la expresión los resultados obtenidos con Eigenfaces y Fisherfaces son mejores que con el algoritmo EGBM según las pruebas en[21]. No obstante, su rendimiento no es del todo malo ya que obtiene un porcentaje de acierto del 91,5%.

En la Tabla VII, se pueden ver los resultados ante cambios en la luminosidad. En este caso, los resultados de EGBM tienen un resultado muy inferior a Eigenfaces y Fisherfaces. Este hecho llama la atención ya que, a priori, las técnicas basadas en modelos son más robustas frente a cambios de este tipo.

TABLA VII.
COMPARATIVA CON OTROS ALGORITMOS EN YALE B CROPPED DATABASE

	Eigenfaces		Fisherfaces		EBGM	
%	acierto	error	acierto	error	acierto	error
Mejor	72,64	27,36	100	0	65,78	34,22

Fig. 19. Comparativa de resultados con técnica basada en modelo (EBGM). Pruebas hechas en la base de datos de Yale B Cropped y resultados medidos en términos de acierto y error porcentuales.

8 CONCLUSIONES

A lo largo de este informe se ha podido ver un estudio de las técnicas de reconocimiento facial más comunes así como una clasificación de las mismas. También se ha visto el funcionamiento de los sistemas de detección y reconocimiento facial y se han descrito los pasos para el diseño y desarrollo de una aplicación de estas características.

Finalmente, se han evaluado los resultados del sistema implementado en una fase de pruebas.

A pesar de que han ido surgiendo diferentes problemas, principalmente a lo largo de las fases de diseño y desarrollo, se han conseguido solventar a tiempo. De esta forma, se ha conseguido una aplicación final funcional que cumple con todos los objetivos que se habían propuesto durante las primeras fases del proyecto.

La aplicación permite detectar y reconocer varios rostros en imágenes. No obstante, si no se trata de imágenes con caras frontales, su rendimiento no está totalmente optimizado y el rendimiento obtenido puede ser inferior.

A nivel personal y como conclusión puedo decir que, en general, la experiencia obtenida tras el desarrollo de este proyecto sido muy positiva y enriquecedora. He aprendido a trabajar de manera autónoma en una área que, a pesar de no estar estrictamente relacionada con la mención cursada, me despertaba bastante interés.

También he aprendido a utilizar nuevas herramientas y algoritmos. No solo he obtenido conocimientos a nivel práctico, también a nivel teórico gracias a la fase previa de documentación y búsqueda de información así como los pasos para desarrollar un proyecto de estas características.

8.1 Líneas futuras

Puesto que en este proyecto no se han podido incluir todas las funcionalidades deseadas, tanto por falta de tiempo como por falta de recursos, se deja abierta la posibilidad de implementar funcionalidades en un futuro.

Una de las principales características a implementar tras la finalización del proyecto es el reconocimiento de rostros en tiempo real. Teniendo implementado el sistema que reconoce rostros a partir de diferentes tipos de archivos de entrada localizados en un directorio, los cambios que se deben hacer para implementar un sistema de reconocimiento facial en tiempo real son relativamente simples.

Otra de las mejoras que se puede añadir para mejorar el proyecto en un futuro, sería llevar a cabo una interfaz

gráfica ya sea para PC o para plataformas móviles (Android, iOS...).

BIBLIOGRAFÍA

- [1] Turk, M., Pentland, A., Eigenfaces for recognition. Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71–86, Mar. 1991.
- [2] Taigman, Y., Yang, M. y Ranzato, M. (2014) DeepFace: Closing the Gap to Human-Level Performance in Face Verification. California: Facebook AI Research.
- [3] Zhang, N., Paluri, M., Taigman, Y., Fergus, R. y Bourdev, L. (2015). Beyond Frontal Faces: Improving Person Recognition Using Multiple Cues. Berkeley: Facebook AI Research.
- [4] F. Schroff, K. Kalenichenko, J. Philbin. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. California: Google.
- [5] A.K. Agrawala y Y.N. Singhb, "Evaluation of face recognition methods in unconstrained environments," Procedia Computer Science, vol. 48, pp. 644–651, 2015.
- [6] Ble, C. Modelo en cascada (2013). http://librosweb.es/libro/tdd/capitulo_1/modelo_en_cascada.html [Consulta: 1 Marzo de 2016]
- [7] Python, <https://www.python.org/> [Consulta: 10 de Abril de 2016]
- [8] OpenCV, docs.opencv.org/3.1.0/ [Consulta: 1 de Mayo de 2016]
- [9] NumPy, www.numpy.org/ [Consulta: 8 de Mayo de 2016]
- [10] Viola and Jones, "Rapid object detection using a boosted cascade of simple features", Computer Vision and Pattern Recognition, 2001.
- [11] ISO/IEC 19794-5:2011. Information technology - biometric data interchange formats - part 5: Face image data. In International Organization for Standardization, 2011
- [12] Histograms - 2: Histogram Equalization, http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_histograms/py_histogram_equalization/py_histogram_equalization.html [Consulta: 29 de Abril de 2016]
- [13] Turk, M., Pentland, A., Eigenfaces for recognition. Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71–86, Mar. 1991.
- [14] Welling, M., Fisher Linear Discriminant Analysis. Department of Computer Science, University of Toronto.
- [15] P. Belhumeur, J. Hespanha, D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," IEEE Transactions on Pattern Analysis and Machine Intelligence, July 1997, pp. 711–720.
- [16] The database of faces, <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html> [Consulta: 5 de Junio de 2016]
- [17] K.C. Lee, J. Ho, D. Kriegman., Acquiring Linear Subspaces for Face Recognition under Variable Lighting. IEEE Trans. Pattern Anal. Mach. Intelligenc, vol. 27, no. 5, pp. 684–698, 2005.
- [18] <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>.
- [19] L. Wiskott, J.-M. Fellous, N. Krueger, C. von der Malsburg, Face Recognition by Elastic Bunch Graph Matching, Chapter 11 in Intelligent Biometric Techniques in Fingerprint and Face Recognition, eds. L.C. Jain et al., CRC Press, 1999, pp. 355–396
- [20] R. Shyam and Y. Narain Singh (2015), Identifying Individuals using Multimodal Face Recognition Techniques, Procedia Computer Science, Elsevier, Vol. 48, pp. 666 – 672.
- [21] F. Bhat & M. Arif Wani., Elastic Bunch Graph Matching Based Face Recognition Under Varying Lighting, Pose, and Expression Conditions. International Journal of Advance Foundation and Research in Science & Engineering (IJAFRSE), Vol. 1, Issue 8, Enero 2015.